Nuclear Burning in Astrophysics

A Broad Introduction

Alexander Smith Clark

PennState

MOTIVATION

- To fully understand astrophysical transients, as CCSN, BNS-mergers, X-burst, and more, we need to include (Mezzacappa, 2026):

  1. A full GR treatment of gravity, hydrodynamics, MHD and neutrino kinetics (AthenaK).
  2. The inclusion of nuclear burning and the proper treatment of stellar matter out of NSE (pynucastro).
  3. The development of a consistent microphysics that provides the right closure and coupling between the hydrodynamic sector and stellar matter.

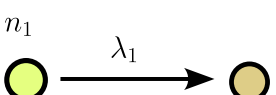
- In this context, we will present developments toward the construction of the reactive sources and to explore the several strategies used to couple them with hydrodynamics.

- Due to several constraints, nuclear burning has been traditionally coupled with Newtonian gravity only: Euler Equations + Reactive Sources (Operator Splitting, Strang, SDC) .

Nuclear Reaction Rates: Definitions

1-Nuclei Decay Rate:

- Let's consider the following situation:



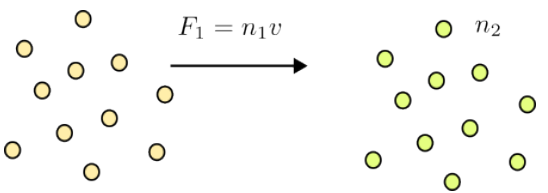
- The change of rate of a photodisintegration/weak decay rate for 1-nuclei is proportional to its present number density:

$$r = -\frac{dn_1}{dt} = \lambda_1 n_1$$

Nuclear Reaction Rates: Definitions

2-Nuclei Scattering Rate:

- Let us consider the following projectiles + target situation:



- Baby nuclear reaction rate:

$$r = -\frac{dn_1}{dt} = -\frac{dn_2}{dt} = F_1\, n_2 \sigma_{12} = n_1 n_2 \sigma(v_{12}) v$$

- After computing the ensemble average in the CM frame:

$$r = \frac{n_1 n_2}{1+\delta_{12}} \int \sigma(v) v P_{M.B}(v^2) d^3 v = \frac{n_1 n_2}{1+\delta_{12}} \langle \sigma v \rangle_{12}$$

Nuclear Reaction Rates: Definitions

2-Nuclei Scattering Rate:

- The value of $\sigma(v)$ is defined by several contributions:

$$\sigma(v) = \sigma_{\mathrm{NR}} + \sigma_{\mathrm{R}} + \mathrm{corrections}$$

- Given the definition of mass fraction (X_i) and abundance (Y_i):

$$X_i = n_i A_i / \rho N_A \qquad Y_i = n_i / \rho N_A$$

  we can write the following identification:

$$\frac{dn_1}{dt} = -\frac{n_1 n_2}{1+\delta_{12}}\langle \sigma v \rangle \rightarrow \frac{dY_1}{dt} = -\frac{\rho Y_1 Y_2}{1+\delta_{12}}[N_A \langle \sigma v \rangle]_{12}$$

- For a generic reaction network, the specific energy transferred from all the reactions that contributes to to the cell ΔY_i is:

$$\boxed{\dot{e}_{nuc} = -N_A c^2 \sum_i m_i \frac{dY_i}{dt}}$$

Ion-Gas Microphysics

- Consider the gran canonical partition function definition:

$$\mathcal{Z} = \sum_k z^k Z_k$$

  where Z_k is the statistical partition function, defined by $Z_k = \frac{1}{k!}(Z_1)^k$.

- For the Ion-Electron Cloud gas, each particle energy is:

$$\epsilon = Wmc^2 + \Delta_\ell + \mu^C \sim \frac{\mathbf{p}^2}{2m} + \Delta_I + mc^2 + \mu^C$$

- The statistical partition function for one particle satisfies:

$$Z_1 = \sum_\ell \int d^3x\, d^3p \exp\left(-\frac{\epsilon}{k_B T}\right)$$

- After computing the Grand-Canonical Function and computing:

$$\Omega = -k_B T \log \mathcal{Z}$$

we compute the total number of particles as a function of μ.

$$\begin{aligned} N &= -\left(\frac{\partial \Omega}{\partial \mu}\right)_{V,T} \\ &= V\left[\sum_l (2J_l + 1) e^{-\Delta_l / k_B T}\right] \left(\frac{m k_B T}{2\pi\hbar^2}\right)^{3/2} e^{(\mu - mc^2 - \mu^C)/k_B T} \end{aligned}$$

- And the chemical potential:

$$\mu = k_B T \log\left[\frac{n}{(2J_0 + 1) G(T)} \left(\frac{2\pi\hbar^2}{m k_B T}\right)^{3/2}\right] + mc^2 + \mu^C$$

Nuclear Screening

- Once the nuclei fully ionizes, the free electrons create a cloud that attenuates a nuclear reaction rates by a factor of $\exp\left(\mu^C/k_BT\right)$, where:

$$\mu^C = \sum_{i=1}^{r} \mu_i^C - \mu_{1+2+\cdots+r}^C$$

. For a reaction with reactants R_1, R_2, $\cdots$, R_r.

- The Coulomb interaction is condensed in terms of the quantity $\mu^C(\Gamma)$, where $\Gamma_i = \beta(Ze)^2/a_i = \Gamma_e Z_i^{5/3}$, and fitted in Chabrier and Potekhin, 1998.

$$F_i^{sc} = \frac{\mu_i^C}{k_BT} = A_1\left[\sqrt{\Gamma_i(A_2+\Gamma_i)} - A_2\ln\left(\sqrt{\frac{\Gamma_i}{A_2}}\sqrt{1+\frac{\Gamma_i}{A_2}}\right)\right] + 2A_3\left[\sqrt{\Gamma_i} - \arctan(\sqrt{\Gamma_i})\right]$$

Reverse Nuclear Rates

No screening | 2+2 nuclei

- Consider a reaction $1 + 2 \rightarrow 3 + 4$. Because the rate at which reactants 1,2 are consumed is the same rate at which the products are created:

$$-\frac{dn_{1,2}}{dt} = \frac{dn_{3,4}}{dt} \Rightarrow \frac{n_1 n_2}{n_3 n_4} = \frac{\langle \sigma v \rangle_{3,4}}{\langle \sigma v \rangle_{1,2}}$$

- Then after plugging back the expression we computed for the number density of each nuclei:

$$\frac{N_A \langle \sigma v \rangle_{3,4}}{N_A \langle \sigma v \rangle_{1,2}} = \frac{(2J_1 + 1)(2J_2 + 1)}{(2J_3 + 1)(2J_4 + 1)} \left(\frac{A_1 A_2}{A_3 A_4} \right)^{3/2} \times \frac{G_1 G_2(T)}{G_3 G_4(T)} \mathrm{e}^{-Q/k_B T}$$

- Usually its customary to assume that $G_i(T) = 1$, but as $T \rightarrow 1\,\mathrm{MeV}$, This assumption is not longer valid.

- In this case, the nuclei disassociate in neutrons and protons.

$$\mu_N = Z\mu_p + N\mu_n$$

After plugging our computed expression for the chemical potential because protons and nuclei are also nuclei embebbed in an alectron cloud:

$$X_i = \frac{m_i}{\rho}(2J_i + 1)G(T)\left(\frac{m_i k_B T}{2\pi\hbar^2}\right)^{3/2} \times \exp\left[\frac{Z_i\mu_p^{\mathrm{id}} + N_i\mu_n^{\mathrm{id}} + Q_i}{k_B T} + \frac{Z_i\mu_p^C - \mu_i^C}{k_B T}\right]$$

- If the electron mass fraction is known:

$$Y_e = \sum_i \frac{Z_i X_i}{A_i}$$

and

$$\sum X_i = 1$$

we can solve for μ_p^{id} and μ_n^{id}.

- Therefore, for each (ρ, T, Y_e) tuple is possible to compute the exact composition in NSE.
- NSE occurs at $T \sim 10^9 K$. However, the exact threshold of density and temperature or reaching mechanism is not known.
- The electron+matter interaction is important. Without it, the absolute error may jump to $\sim 10^{-3}$.

Nuclear Reaction Libraries

REACLIB

- Main reference: https://reaclib.jinaweb.org/)
- This library tabulates each nuclear reaction rate by several contributions of the following data fitting $\{a_i\}_{i=0}^{6}$:

$$N_A\langle\sigma v\rangle = \exp\left(a_0 + \sum_{i=1}^{5} a_i T_9^{(2i-5)/3} + a_6 \log T_9\right)$$

- An example of a reaclib rate:

```
2
sc41    p ca40                        il10nv   -1.08509e+00
4.159690e+01-1.259250e+01-3.117110e+01 0.000000e+00
0.000000e+00-1.000000e+00 8.333330e-01
```

- Main reference: https://research.physics.unc.edu/nuclearastro/Starlib.html
- This library tabulates the necessary parameters to reconstruct its log-normal distribution.

$$P(x;\mu;\sigma) = \frac{1}{\sqrt{2\pi}\sigma}\frac{1}{x}\exp\left(-(\log(x)-\mu)/2\sigma^2\right)$$

- An extract of an starlib rate:

```
1        n    p                                   be12w      7.82347e-01
1.00E-03     1.136E-03     1.001E+00
2.00E-03     1.136E-03     1.001E+00
3.00E-03     1.136E-03     1.001E+00
4.00E-03     1.136E-03     1.001E+00
5.00E-03     1.136E-03     1.001E+00
6.00E-03     1.136E-03     1.001E+00
```

- Let us recall the previous table:

```
1        n    p                                 be12w     7.82347e-01
1.00E-03     1.136E-03     1.001E+00
2.00E-03     1.136E-03     1.001E+00
3.00E-03     1.136E-03     1.001E+00
4.00E-03     1.136E-03     1.001E+00
5.00E-03     1.136E-03     1.001E+00
6.00E-03     1.136E-03     1.001E+00
```

- The columns correspond to temperature (K), median value ($x_{\rm med}$) and the factor uncertainty (f_u) respectively:

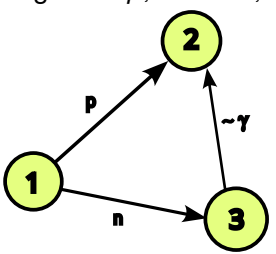
$$x_{\rm low} = x_{\rm med}/f_u, \qquad x_{\rm high} = x_{med} f_u$$

- The STARLIB Library can manage error propagation!!.
- REACLIB does not include the nuclear partition functions. ($G_i(T) = 1$)
- REACLIB reverse rates are pretty simple!!.

Nuclear Reaction Networks

Formulate the Nuclear Reaction Network

- Consider three nuclei 1,2 and 3, and background p, n nuclei, fixed at some (ρ, T) tuple:



- From here, we may construct the overall reaction network.

$$\frac{dY_1}{dt} = -Y_1 Y_p[\rho N_A \langle \sigma v \rangle_{1p}] - Y_1 Y_n[\rho N_A \langle \sigma v \rangle_{1n}]$$
$$\frac{dY_2}{dt} = +Y_1 Y_p[\rho N_A \langle \sigma v \rangle_{1p}] + Y_3 \lambda_3$$
$$\frac{dY_3}{dt} = +Y_1 Y_n[\rho N_A \langle \sigma v \rangle_{1n}] - Y_3 \lambda_3$$
$$\frac{dY_n}{dt} = -Y_1 Y_n[\rho N_A \langle \sigma v \rangle_{1n}]$$
$$\frac{dY_p}{dt} = -Y_1 Y_p[\rho N_A \langle \sigma v \rangle_{1p}]$$

Blackboard!

Nuclear Reaction Networks

A semi-realistic case

- Let us solve a reaction network composed of four nuclei: p, $^{12}\mathrm{C}$, $^{13}\mathrm{N}$, and $^{13}\mathrm{C}$ fixed at $(\rho, T) = (10^4\,\mathrm{g/cm^3}, 9.0 \times 10^7\,\mathrm{K})$. How we formulate the reaction network and solve it, given an initial condition?

- Let's explore by pen & paper (Just once to learn!).

- We start by identifying their important rates $^{12}\mathrm{C}(p,\gamma)^{13}\mathrm{N}$ and $^{13}\mathrm{N}(,\mathrm{e}^+\nu_e)^{13}C$.
- We formulate the system:

$$
\begin{aligned}
\frac{dY_{^{12}\mathrm{C}}}{dt} &= -\rho Y_{^{12}C} Y_p N_A \langle \sigma v \rangle_{^{12}\mathrm{C},p} \\
\frac{dY_{^{13}\mathrm{N}}}{dt} &= \rho Y_{^{12}C} Y_p N_A \langle \sigma v \rangle_{^{12}\mathrm{C},p} - \lambda_{^{13}\mathrm{N}} Y_{^{13}\mathrm{N}} \\
\frac{dY_p}{dt} &= -\rho Y_{^{12}C} Y_p N_A \langle \sigma v \rangle_{^{12}\mathrm{C},p} \\
\frac{dY_{^{13}\mathrm{C}}}{dt} &= \lambda_{^{13}\mathrm{N}} Y_{^{13}\mathrm{N}}
\end{aligned}
$$

- Note that $\sum_i A_i \dot{Y}_i = 0$. Is important to check this condition in order to ensure that the mass fraction normalization condition maintains.

- This is condensed in the following notation:

$$\frac{d\mathbf{Y}}{dt} = f(\mathbf{Y})$$

- From here, we can write the following finite difference:

$$\frac{\mathbf{Y}^{n+1} - \mathbf{Y}^{n}}{dt} = f(\mathbf{Y}^{n+1}) \qquad \text{(Backward Euler)} \tag{1}$$

- But, how we evaluate $\mathbf{Y}^{n+1}$? We take a guess on $\mathbf{Y}^{n+1}$ (Let us say $\mathbf{Y}_0$) and expand around it: $\mathbf{Y}^{n+1} \sim \mathbf{Y}_0^{n+1} + \Delta\mathbf{Y}_0$. Thus:

$$f(\mathbf{Y^{n+1}}) \sim f(\mathbf{Y_0^{n+1}} + \acute{}\mathbf{Y_0}) \sim f(\mathbf{Y_0^{n+1}}) + \left.\frac{\partial f}{\partial \mathbf{Y}}\right|_{\mathbf{Y}_0^{n+1}} \Delta\mathbf{Y}_0$$

- Therefore, its possible to write:

$$\frac{\mathbf{Y}^{n+1} - \mathbf{Y}^{n}}{dt} = f(\mathbf{Y}_0^{n+1}) + \underbrace{\left.\frac{\partial f}{\partial \mathbf{Y}}\right|_{\mathbf{Y}_0^{n+1}}}_{\mathbf{J}_0^{n+1}} \Delta\mathbf{Y}_0$$

- After reordering everything, we can write (1)

$$\mathbf{Y}^{n+1} = \mathbf{Y}^n + dt\ \left[f(\mathbf{Y}_0) + \mathbf{J}(\mathbf{Y}_0^{n+1}) \cdot \Delta\mathbf{Y}_0\right] \tag{2}$$

- Or in terms of the correction:

$$(1 - \mathbf{J}_0^{n+1})\Delta\mathbf{Y}_0 = \mathbf{Y}^n - \mathbf{Y}_0^{n+1} + dt\, f(\mathbf{Y}_0^{n+1}) \tag{3}$$

- Finally the following iterative process reveals:

1. We take a guess $\mathbf{Y}_0^{n+1}$, and compute the correction $\Delta\mathbf{Y}_0^{n+1}$ from (18). If $||\Delta\mathbf{Y}_0|| < \epsilon||\mathbf{Y}^n||$ for a given ϵ. The solution converges and the iteration stops.
2. We compute the new guess from (2).
3. The quantity $\mathbf{J}_0^{n+1}$ is the Jacobian matrix evaluated at $\mathbf{Y}_0^{n+1}$.

- The Jacobian for our test case is:

$$\mathbf{J}(\mathbf{Y}) = \begin{bmatrix} -\rho Y_p N_A \langle \sigma v \rangle_{^{12}\mathrm{C},p} & 0 & -Y_{^{12}\mathrm{C}} N_A \langle \sigma v \rangle_{^{12}\mathrm{C},p} & 0 \\ \rho Y_p N_A \langle \sigma v \rangle_{^{12}\mathrm{C},p} & -\lambda_{^{13}\mathrm{N}} & Y_{^{12}\mathrm{C}} N_A \langle \sigma v \rangle_{^{12}\mathrm{C},p} & 0 \\ -\rho Y_p N_A \langle \sigma v \rangle_{^{12}\mathrm{C},p} & 0 & -Y_{^{12}\mathrm{C}} N_A \langle \sigma v \rangle_{^{12}\mathrm{C},p} & 0 \\ 0 & \lambda_{^{13}\mathrm{N}} & 0 & 0 \end{bmatrix}$$

- If we evaluate our Jacobian for our thermodynamics $(\rho, T) = (10^4\ \mathrm{g/cm^3}, 9.0 \times 10^7\ \mathrm{K})$, for reaclib rates (and a bit of cheating!) and an initial condition $X_p = 0.5$, $X_{^{12}\mathrm{C}} = 0.5$:

$$\mathbf{J}(\mathbf{Y}) = \begin{bmatrix} -0.04769791 & 0.0 & -0.00397483 & 0.0 \\ 0.04769791 & -0.00115911 & 0.00397483 & 0.0 \\ -0.04769791 & 0 & -0.00397483 & 0.0 \\ 0 & 0.00115911 & 0 & 0.0 \end{bmatrix}$$

- If we compute the non-zero eigenvalues of this matrix: $\lambda_1 = -0.0516727$, $\lambda_2 = -0.00115911$. Note that $|\lambda_1/\lambda_2| = 44.5796343746$.

- The process of assembling $f(\mathbf{U})$ is tedious, but the construction is simple if we know the nuclei that constitute the network a priori.

- Because, $|\lambda|_{\max}/|\lambda|_{\min} = 44.5796343746 \gg 1$, the ODE system is stiff and an implicit-unconditionally-stable-convergent method may be the best option to secure a meaningful numerical solution.

- Although we describe the linearized Back-Euler method, the Jacobian is used in several implicit numerical schemes. The memory footprint of the Jacobian per MeshBlock (mb), is given by :

$$\sigma \sim N^{x}_{cell/mb} \times N^{y}_{cell/mb} \times N^{z}_{cell/mb} \times N^{2}_{\text{species}} \times (8\,\text{byte})/(1024^2\,\text{byte/MB})$$

.

- For a meshblock with 64 number of cells at each direction, and 50 species:

$$\sigma \sim 4.88\,\text{GB/mb}$$

- Hence, the integration of nuclear reaction networks, has become feasible under GPU-offloading.

PYNUCASTRO

A PYTHON LIBRARY FOR NUCLEAR ASTROPHYSICS

- Repository: https://github.com/pynucastro/pynucastro
- We need a software package that automatizes the construction of the term $f(\mathbf{Y})$.
- We would like to explore elements, like the Jacobian at different values of $(\rho, T, \mathbf{Y})$ from the network
- We would like to integrate a reaction network $f(\mathbf{Y})$ given a set of initial condition.
- It would be interesting to interact and visualize the reaction network, according to our needs.

PYNUCASTRO: NETWORK VISUALIZATION

- Let us consider our previous example: A set of nuclei, p, $^{12}\mathrm{C}$, $^{13}\mathrm{N}$, and $^{13}\mathrm{C}$ fixed at $(\rho, T) = (10^4\,\mathrm{g/cm^3}, 9.0 \times 10^7\,\mathrm{K})$

- Let us consider the following piece of python code:

```
from pynucastro import Library
from pynucastro import PythonNetwork
from pynucastro import ReacLibLibrary
from pynucastro import Composition
```

- From here, we can load all the rates contained in the reaclib library:

```
lib = ReacLibLibrary()
```

- To find all the rates that connect the set of nuclei:

```
nuc = ['p', 'c12', 'n13', 'c13']
rates = lib.linking_nuclei(nuclist=nuc, with_reverse=False)
```

- The `rates` variable is a container for the two reaction rates that coincide with the previous reaction network.
- Let us now create a `PythonNetwork` object:

```
net = PythonNetwork(rates=rates.get_rates())
```

  and a `Composition` object:

```
comp = Composition(nuclei=rates.get_nuclei())
comp.set_array([0.5, 0.0, 0.0, 0.5])
```

- Finally, let us define the density and temperature tuple (ρ, T)

```
rho = 1.0e4
temp = 9.0e7
```

- From here we can plot our network by:

```
net.plot(rho=rho, T=temp, comp=comp, hide_xalpha=True, hide_xp=True,
         node_size=700, node_font_size=10,
         size=(500, 500), outfile='2_spec.pdf')
```

- This generates the following picture:

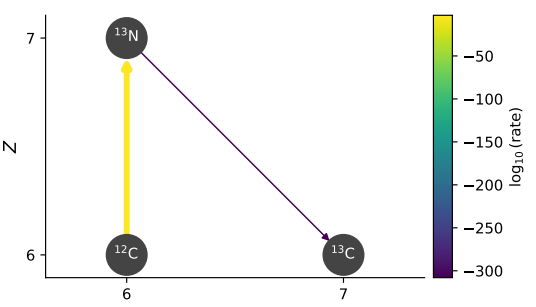


Figure. A reaction network constructed from $^{12}\mathrm{C}(p,\gamma)^{13}\mathrm{N}$ and $^{13}\mathrm{N}(,\mathrm{e}^{+}\nu_e)^{13}C$.

PYNUCASTRO: NETWORK VISUALIZATION

PLOTTING THE JACOBIAN

- From the following option:

```
net.plot_jacobian(rho, temp, comp, rate_scaling=1.e8, outfile='jac.pdf')
```

which produces:

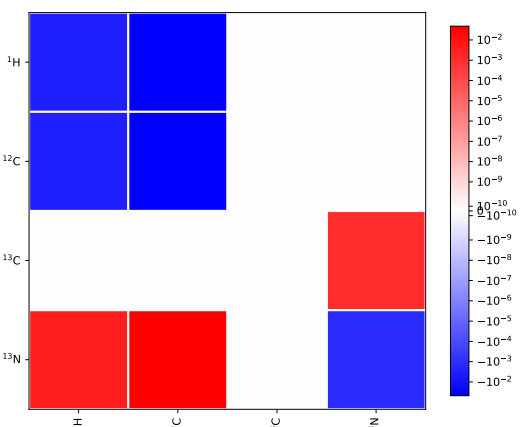


Figure. An ilustration of the Jacobian evaluated at $\rho = 10^4\ \mathrm{g/cm^3}$, and $T = 9.0 \times 10^7\ \mathrm{K}$.

PYNUCASTRO: INTEGRATING THE NETWORK

- Now, we write our network by:

```
net.write_network('net_2.py')
```

- The module `net_2.py` contains all the rate definitions, partition functions for detailed balance reversed rates, the atomic mass number, screening factors and all the necessary information of $f(\mathbf{U})$ and Jacobian $\mathbf{J}$.

  Let us now integrate it by creating the following initial condition:

```
import net_2
X0 = np.zeros(net_2.nnuc)
X0[net_2.jp] = 0.5
X0[net_2.jc12] = 0.5
X0[net_2.jn13] = 0.0
X0[net_2.jc13] = 0.0
Y0 = X0/net_2.A
```

PYNUCASTRO: INTEGRATING THE NETWORK

- Let us now integrate the network using an implicit ODE solver from the `sympy` package:

```
tmin = 1.0e-4
tmax = 1.0e6
from scipy.integrate import solve_ivp
from pynucastro.screening import potekhin_1998
sol =  solve_ivp(net_2.rhs, [0, tmax], Y0, method="BDF",
                 dense_output=True, args=(rho, T, potekhin_1998),
                 rtol=1.e-10, atol=1.e-10, jac=net_2.jacobian)
```

- Note that we added the nuclear screening effect in the integration of our network.

PYNUCASTRO: INTEGRATING THE NETWORK

- After plotting `sol` we obtain the following evolution from our example network.

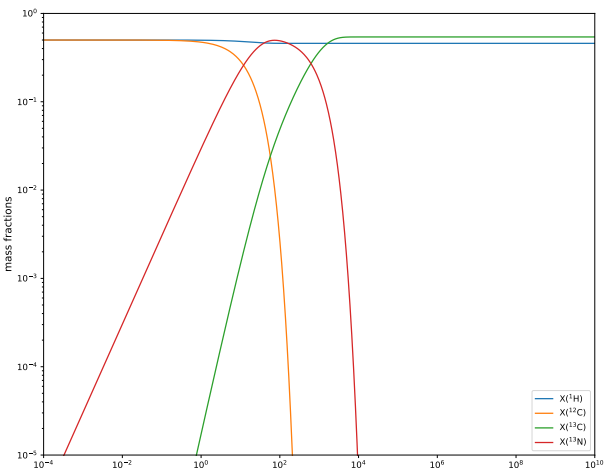


Figure. The solution of our previously defined network

Analysis Reaction Networks

Cold CNO vs Hot CNO

- We will start comparing the Cold-CNO vs Hot-CNO for a density of $\rho = 10^4\,\mathrm{g/cm^3}$ and $T_c = 9.0 \times 10^7\,\mathrm{K}$, $T_h = 5.0 \times 10^8\,\mathrm{K}$.

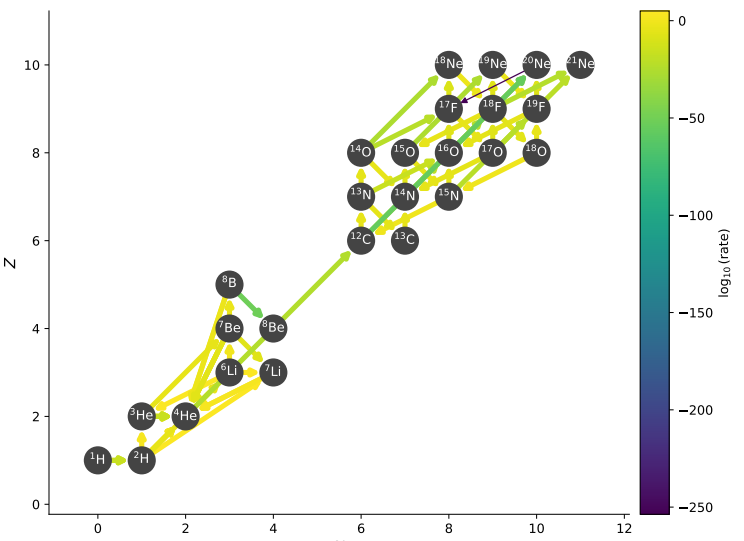


Figure. Cold-CNO

Analysis Reaction Networks

Cold CNO vs Hot CNO

- Now if we increase the temperature a little bit!:

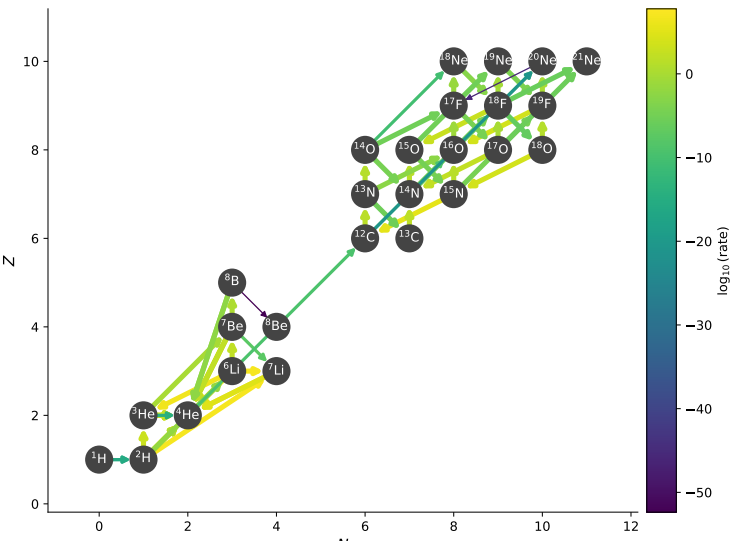


Figure. Hot-CNO

EXAMPLE: NOVAE OUTBURST

- The transition between cold and hot CNO, is of significant important in the description of novae outbursts:

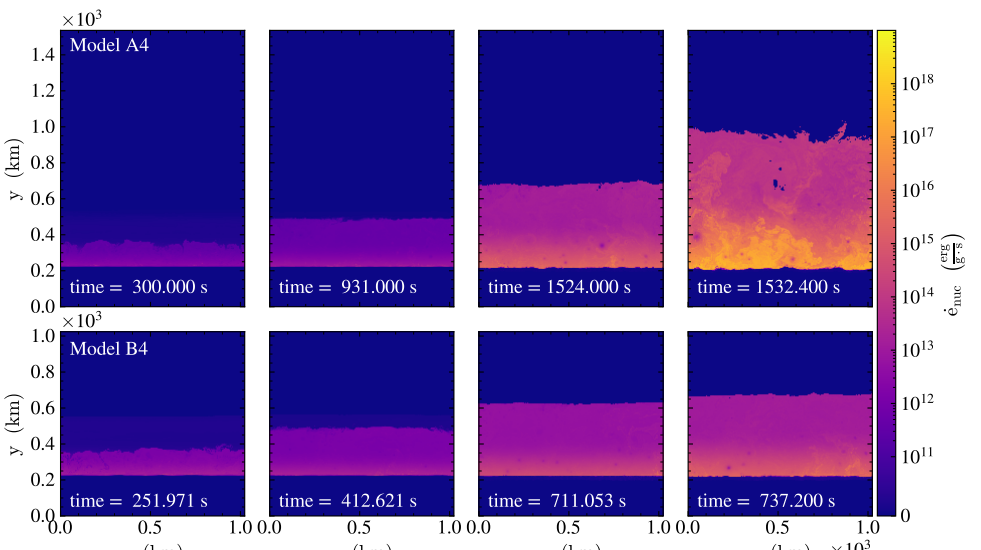


Figure. Novae Outburst

- The whole classic NR-hydro system:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} &= -\nabla \cdot (\rho \mathbf{v}) \\
\frac{\partial (\rho \mathbf{v})}{\partial t} &= -\nabla \cdot (\rho \mathbf{v}\mathbf{v} + p) + \rho \mathbf{g} \\
\frac{\partial (\rho E)}{\partial t} &= -\nabla \cdot (\rho \mathbf{v} E + p \mathbf{v}) + \nabla \cdot (\mathcal{K}_{\mathrm{th}} \nabla T) + \dot{\epsilon}_{\mathrm{nuc}} \\
\frac{\partial (\rho X_{n_i})}{\partial t} &= -\nabla \cdot (\rho \mathbf{v} X_{n_i}) + \rho \dot{\omega}_{n_i}
\end{aligned}$$

$$\underbrace{\frac{\partial \mathcal{U}}{\partial t}} \qquad \underbrace{\mathcal{A}} \qquad\qquad \mathcal{S} = \overbrace{\mathbf{H}(\mathcal{U}) + R(\mathcal{U})}$$

- A coupling strategy is necessary to alternate between advection and reactive sources. The default choice is the operator splitting operator.

- The generalization to GR-hydro:

$$\begin{aligned} \nabla_\mu(\rho u^\mu) &= 0 \\ \nabla_\nu T^{\mu\nu} &= 0 \\ \nabla_\mu(\rho X_i u^\mu) &= A_i m_u \dot{n}_{\rm nuc} \end{aligned}$$

- It would definitely be very interesting to port pynucastro in AthenaK!!
- Nuclear network integration in GR-Hydro context!!